

WoT Thing Description Protocol Binding Template for OCF

Michael J Koster

June 8, 2017

Protocol Binding

- Maps abstract operations on WoT meta types to concrete operations on target resources
- WoT
 - Properties (get, set)
 - Actions (invoke, cancel)
 - Events (subscribe, unsubscribe)
- OCF
 - CRUD+N Resource Model, resource types, interface types, collections, actuation

Protocol Binding Template

- A specification for the information included in a particular protocol binding, and its format
- TD Interaction Description contains information common to all protocol bindings
- Protocol Binding contains information needed to make requests and process responses
- Protocol Binding Template indicates which information is included for a particular protocol

Thing Description (simplified)

```
{
  "semtype": ["thing", "sch:light"],
  "name": "Example Light Thing",
  "interactions": [
    {
      "semtype": ["action", "sch:setlevel"];
      "name": "set brightness level",
      "inputdata": {
        "type": "number"
        "semtype": "sch:level"
      },
      "links": [
        { "href": "/example/light" },
        {...}
      ]
    },
  ]
}
```

Interaction
Description

Protocol
Binding

OCF Protocol Binding Structure

- Basic form of a hypermedia control, extension to the "links" property
 - link attributes describe high level operations
 - method, interface type, resource type, schema mapping to input/output data in TD
- Properties included in the OCF Protocol Binding
 - method (ocf.retrieve, ocf.update...)
 - rt - resource type
 - if - interface type
 - mediatype (default "application/vnd.ocf+cbor")
 - schemas for inputdata/outputdata element

TD – Interaction Description

JSON Schema
+ Extension

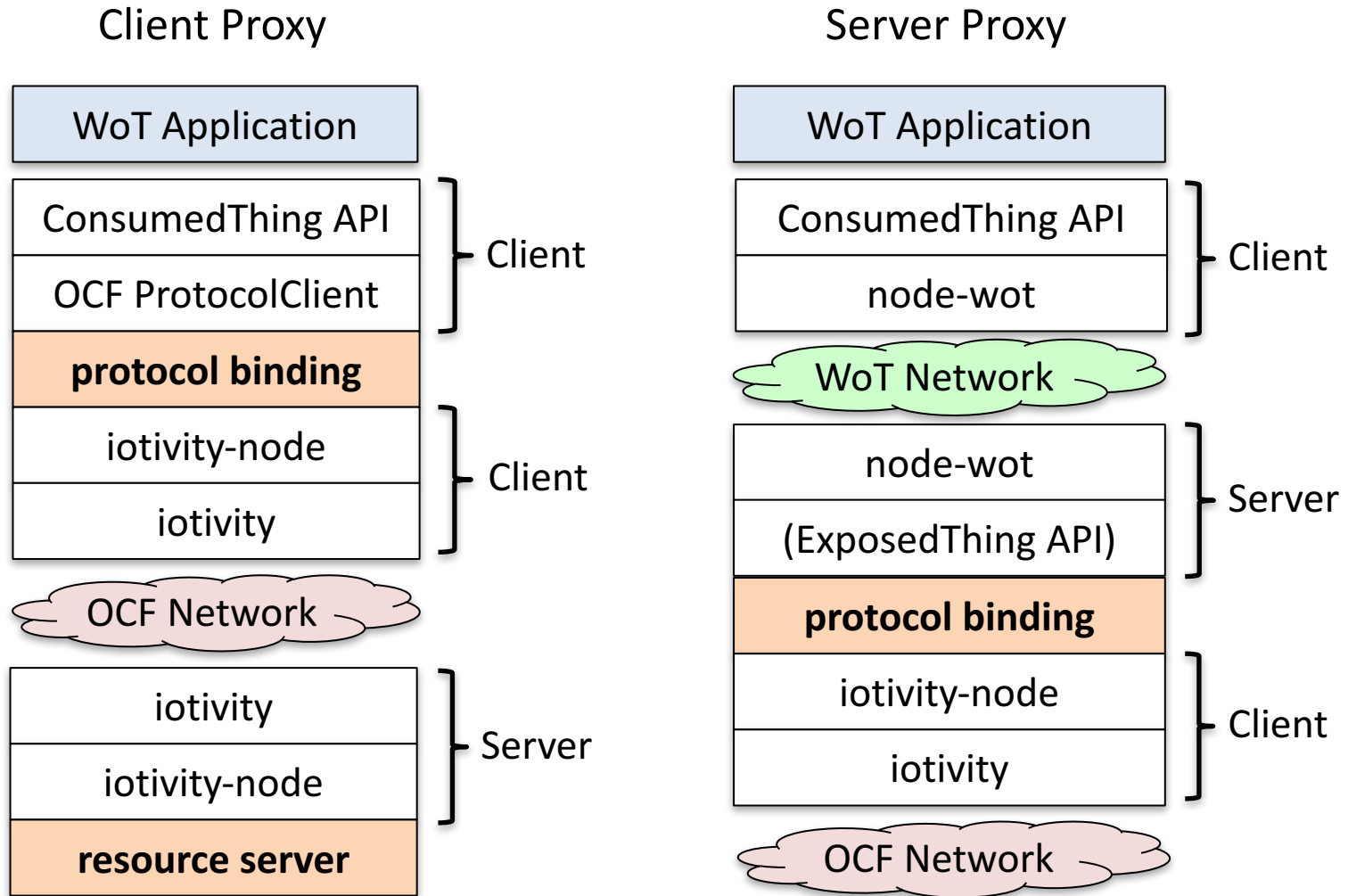
```
{
  "@type": ["action", "sch:setlevel"];
  "name": "set brightness level",
  "inputdata": {
    "type": "object"
    "properties": {
      "P0001": {
        "type": "number",
        "@type": "sch:level.value"
      },
      "P0002": {
        "type": "number",
        "@type": "sch:transitiontime.value"
      }
    }
  },
  "links": [
    (protocol binding goes here)
  ]
}
```

TD - Protocol Binding

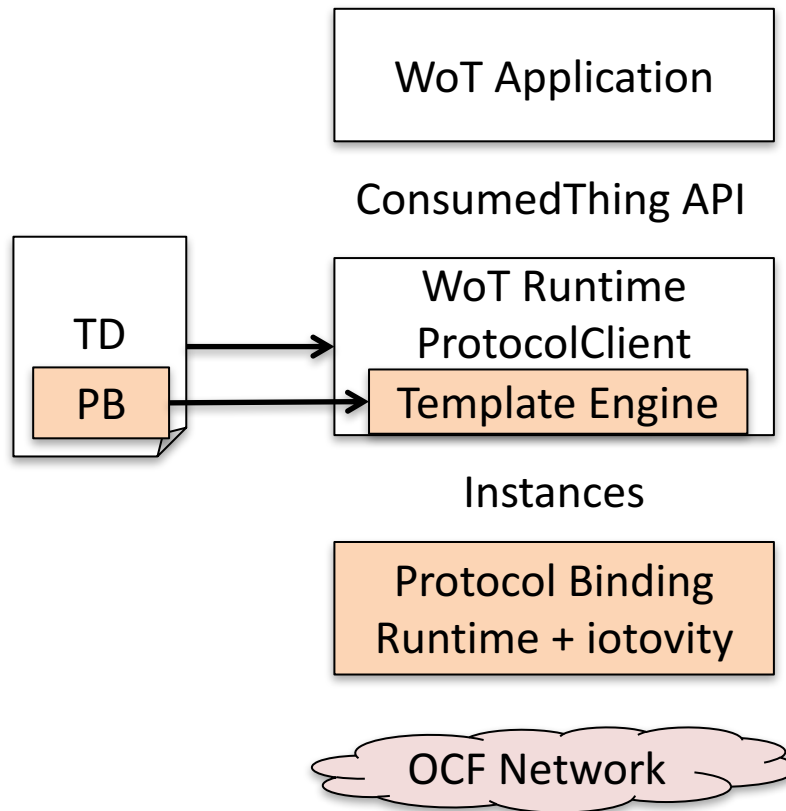
```
links: [  
  {  
    "href": "coaps://[fdfd::22]/example/light/bright",  
    "content-format": 1000,  
    "method.invoke": "post",  
    "rt": ["oic.r.brightness", "oic.r.ramptime"],  
    "if": ["oic.if.a"],  
    "inputschema": {  
      "type": "object",  
      "properties": {  
        "brightness": {  
          "value": "{{P0001}}"  
        },  
        "ramptime": {  
          "value": "{{P0002}}"  
        }  
      }  
    }  
  }  
]
```

JSON Schema
+ Extension

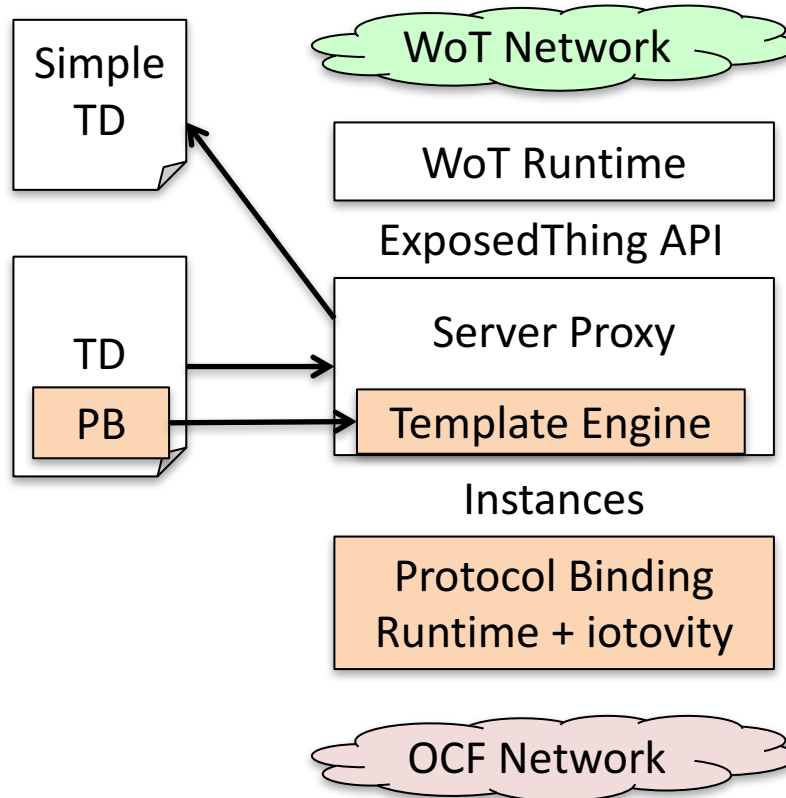
Integration Patterns



Protocol Binding Processing for Client Proxy Pattern



Protocol Binding Processing for Server Proxy Pattern



Discovery

- Applications discover things using TD templates containing semantic constraints
- TD instances could be pre-generated for all OCF discovered devices, and registered with a TD discovery service
- New devices could be registered upon OCF discovery
- Alternatively, OCF discovery queries could be mapped from Thing Descriptions and submitted on demand

Mapping OCF to WoT

- OCF Device maps to Wot Thing
- An OCF device contains a set of resource type instances
- An OCF resource type instance contains a set of properties
- Multiple properties may be retrieved or updated in a single interaction, e.g. a value and a time stamp
- OCF properties map to TD properties
- Some OCF properties map to TD property value attributes (min, max, step, units)

Mapping OCF to TD Actions

- With OCF, Actions are invoked using updates to properties
- Multiple properties may be updated in a request, for example light brightness and ramp time, by sending a structure
- OCF does not define actions per se

Capability-based mapping

- Capabilities are Action, Event, and Property definitions for a single function
- On/off control is a capability with an on/off boolean state property, and either an on action and an off action, or a state change action that has a boolean input parameter
- Level control is a capability with level and transition time properties, and a setlevel action with level and transitiontime input parameters

Capability-based mapping

- An OCF light exposes a binary switch resource, a brightness control resource, and a ramp time resource
- OnOff and level capabilities can be used to map the binary switch and brightness + ramp time resources, respectively
- TD exposes the properties and actions for onoff and level, and maps them to the OCF resources and methods

Examples

Examples in this presentation may be found in:

<https://github.com/mjkoster/wot-protocol-binding>